

THE WESTERN AERONAUTICAL TEST RANGE

CHAPTER 10 TOOLS

Kevin Knudtson, Alice Park
NASA Dryden Flight Research Center, Edwards, California

Bob Downing, Jack Sheldon, Robert Harvey, April Norcross
Arcata Associates, Incorporated, Edwards, California

ABSTRACT

The Western Aeronautical Test Range (WATR) staff at the NASA Dryden Flight Research Center is developing a translation software called *Chapter 10 Tools* in response to challenges posed by post-flight processing data files originating from various on-board digital recorders that follow the Range Commanders Council Inter-Range Instrumentation Group (IRIG) 106 Chapter 10 Digital Recording Standard but use differing interpretations of the Standard. The software will read the data files regardless of the vendor implementation of the source recorder, displaying data, identifying and correcting errors, and producing a data file that can be successfully processed post-flight.

Keywords: *Range Commanders Council, Inter-Range Instrumentation Group (IRIG) 106 Chapter 10 Digital Recording Standard, software tool, Telemetry Attributes Transfer Standard (TMATS), pulse code modulation (PCM)*

INTRODUCTION

The National Aeronautics and Space Administration (NASA) Dryden Flight Research Center (DFRC) (Edwards, California) Western Aeronautical Test Range (WATR) supports concurrent flight-testing across multiple flight research projects. As part of the DFRC flight research environment, flight data is converted to a DFRC standard processed data format called CMP4, which is a compressed-file format containing descriptive information about the contents of the file. Converting data files that are produced following the Range Commanders Council Inter-Range Instrumentation Group (IRIG) 106 Chapter 10 Digital Recording Standard, or “Chapter 10” data files, requires the use of commercially-available software to read the files. In-house-developed software is then used to convert the raw data output from the Chapter 10 reader software to engineering units processed data in the DFRC CMP4 file format. These CMP4 files are then transferred to a central storage system and are available to flight researchers across the DFRC campus.

THE WESTERN AERONAUTICAL TEST RANGE

CHAPTER 10 RECORDER EXPERIENCE

The WATR began to see the use of airborne Chapter 10 recorders approximately four years ago. The implicit assumption made by the purchasers when they selected these recorders was that all Chapter 10 recorders would produce a Chapter 10 file compatible with those produced by any other Chapter 10 recorder. In fact, many Chapter 10 recorder manufacturers were promoting this idea.

Numerous problems, however, were encountered while attempting to read and process the various Chapter 10 data files. Occasionally, the software used to read the files would indicate an error but would not indicate what type of error had occurred. The WATR used commercially-available Chapter 10 validation software in an attempt to identify possible problems with the files, but the software application would not always identify a problem. Sometimes, the validation software would not identify all Chapter 10 problems within a file, and at other times it was difficult to determine the cause of the validation errors. These problems led to the realization that the implementation of the Chapter 10 Standard was subject to interpretation, and that inconsistencies between vendors, or even between one vendor's different firmware versions of the same recorder, could prevent the accurate processing of Chapter 10 data.

An examination of the Chapter 10 files was performed, comparing them to the Chapter 9 Telemetry Attributes Transfer Standard (TMATS) and the Chapter 10 Standard, in order to establish what would be considered actual errors from what would be considered different interpretations of the Chapter 10 Standard. Some types of errors were seen consistently across multiple manufacturers. Time errors (time jumps, static time, and time reversal) were the most common problems identified. Additionally, it appears that most Chapter 10 recorders have difficulty meeting the one-second time stamp requirement for writing the time packets to the file.

Some of the errors that were observed in the TMATS section of the Chapter 10 data files included attribute values that exceeded the maximum limit, the inclusion of random characters, slight variations in the naming of attributes, and invalid attribute values. It was also concluded that although the validation software extracted some information from the TMATS section, it does not appear to perform validation on any of it.

THE NEED FOR A SOLUTION

Initially, files that could not be processed were returned to the originating project team with the encountered errors identified. The team, however, had no method of resolving these problems and subsequently sent the files back to the WATR. The WATR staff then manually edited some of these files to repair problems, but determined that this method was not a viable long-term solution due to the size of the files and the complexity of re-computing the Chapter 10 header structures and checksums. A significant amount of time was spent trying to distinguish between actual errors and what were merely differences in implementing the Chapter 10 Standard, but no clear answers could be found. It seemed reasonable to believe that the proliferation of Chapter

10 recorders would continue and that it would be necessary to find a more reasonable solution to support our customers. A software tool was needed that could take any Chapter 10 file and convert it to a Chapter 10 file that could successfully be processed by the WATR post-flight data processing systems.

Based on discussions with the user community to promote cooperation between vendors and development of software for analyzing data from a Chapter 10 recorder, it was concluded that there was currently no tool available to meet the needs of the WATR staff. After considering many of the potential solutions available, it was decided that an in-house development would be the most cost-effective long-term approach.

The *Chapter 10 Tools* is a multipurpose software application written in Visual C#® using Visual Studio 2008® (both of Microsoft Corporation, Redmond, Washington). The application will provide the user with the ability to validate the Chapter 10 file, edit its contents, and automatically correct errors whenever possible. A packet viewer allows visibility into the packets data, and a TMATS editor is also available to modify and add details to the TMATS file.

APPLICATION ARCHITECTURE

A modular software design with a plug-in architecture was chosen as the most effective approach because of the anticipated developer resource constraints and Chapter 10 specification revisions. This approach eliminates the need to develop the full functionality of the application from the outset; it also allows for an incremental approach in meeting current and future requirements. The plug-in architecture was used for the parts of the application referred to as the Data Type Modules. The Data Type Modules detect and correct errors in Chapter 10 data packets. The data types supported by the Chapter 10 Standard include Time, Pulse Code Modulation (PCM), MIL-STD-1553 bus, ARINC-429 bus, Video, Ethernet, Analog, Discrete, Computer Generated, Message, Image, UART, IEEE 1394 and Parallel. The initial release of the *Chapter 10 Tools* scheduled for Fall 2011 will implement the following data types: Computer Generated, Time, and PCM. Subsequent releases will add MIL-STD-1553, ARINC 429, and Ethernet data types.

The *Chapter 10 Tools* architecture is segmented into five functional areas: the Chapter 10 Reader, the TMATS module, the Controller, the Channel Pipelines, and the Chapter 10 Writer. The *Chapter 10 Tools* architecture is shown in Figure 1.

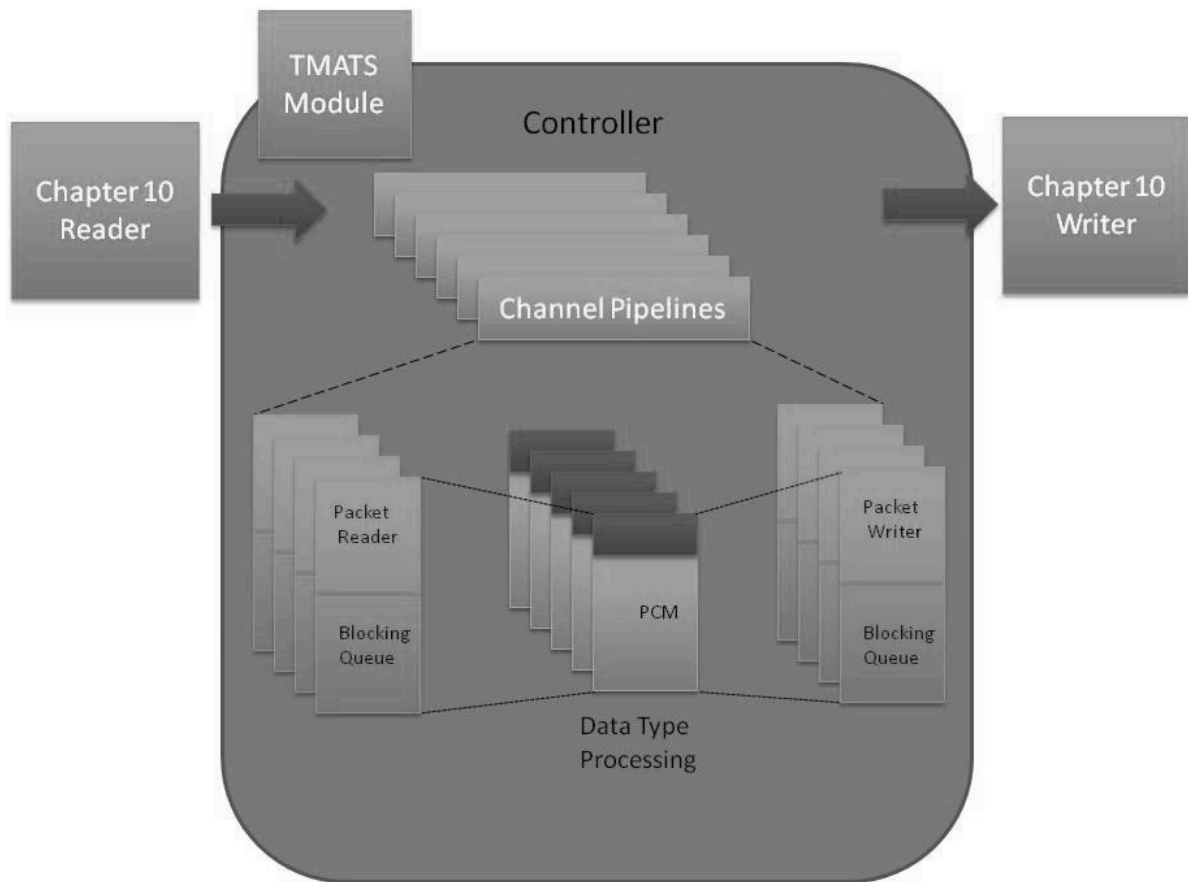


Figure 1. The *Chapter 10 Tools* software architecture.

The Chapter 10 Reader:

- Reads data from the original Chapter 10 data file
- Detects and corrects errors in the Chapter 10 packets
- Strips the Data Type body from the packets and passes them to the Channel Pipelines.

The TMATS Module:

- Detects and corrects errors in the TMATS section
- Provides users the ability to manually correct errors
- Provides the corrected TMATS information to the Controller.

The Controller:

- Receives setup information from the user (channels to be processed, start times, end times, type of error correction to be performed, et cetera)
- Configures the Channel Pipelines
- Passes information necessary for the correct processing of the data to the Data Type modules.

About the Channel Pipelines:

- Each data type has its own pipeline

- Each pipeline has a packet reader/writer and blocking queues to control the flow of data
- Each data type module detects and corrects errors.

The Chapter 10 Writer:

- Gathers the packet bodies from all of the Channel Pipelines
- Assembles the data into a valid Chapter 10 packet format before writing out to the new Chapter 10 file.

CHAPTER 10 PACKET VIEWER

The Chapter 10 Packet Viewer allows the user to view packet data based on a channel selection. The Packet Viewer Graphical User Interface (GUI) provides two options for data display: raw hex format, as shown in Figure 2, or formatted, as shown in Figure 3.

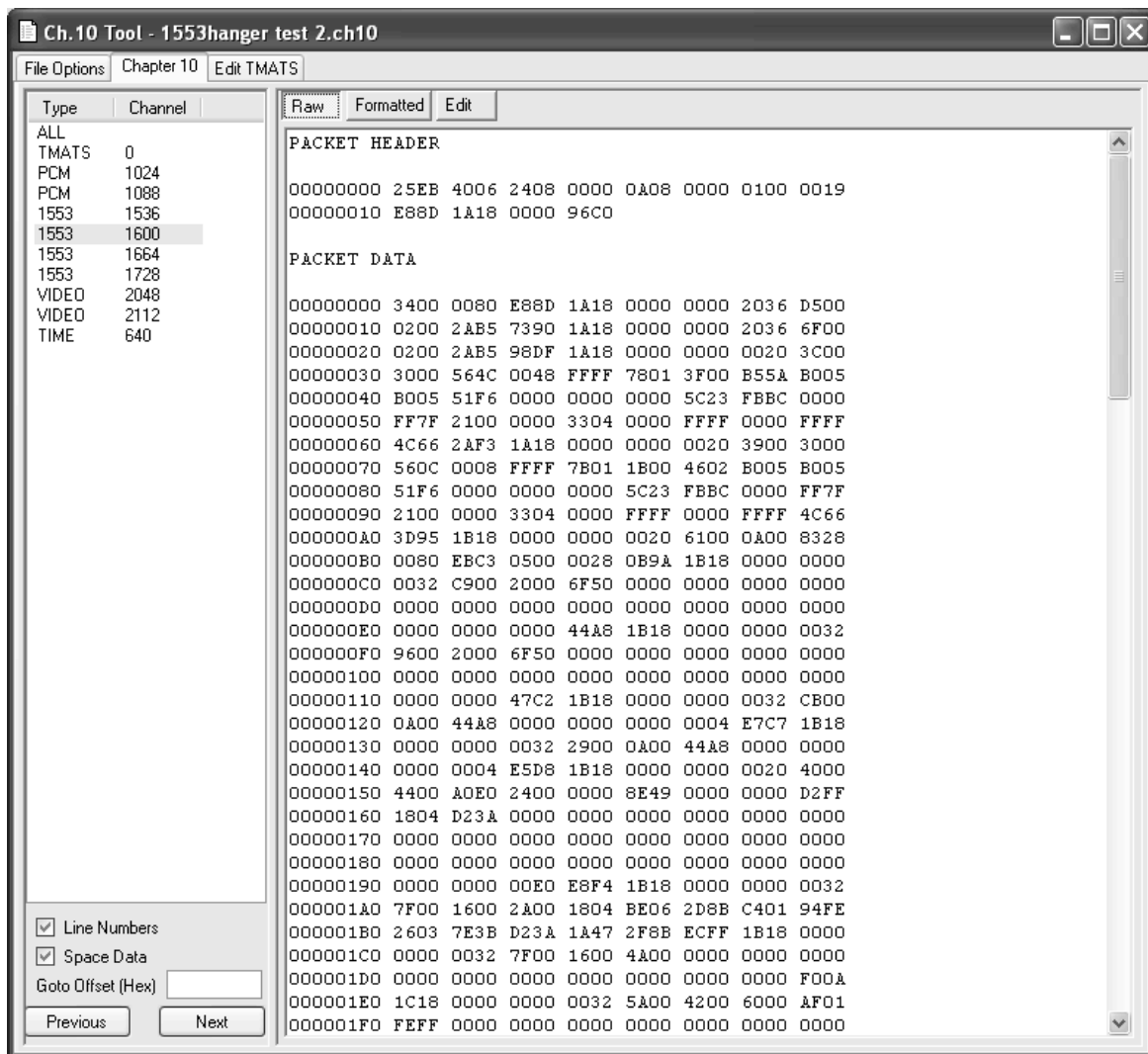


Figure 2. An example of the Chapter 10 Tool Packet Viewer Graphical User Interface: Raw data.

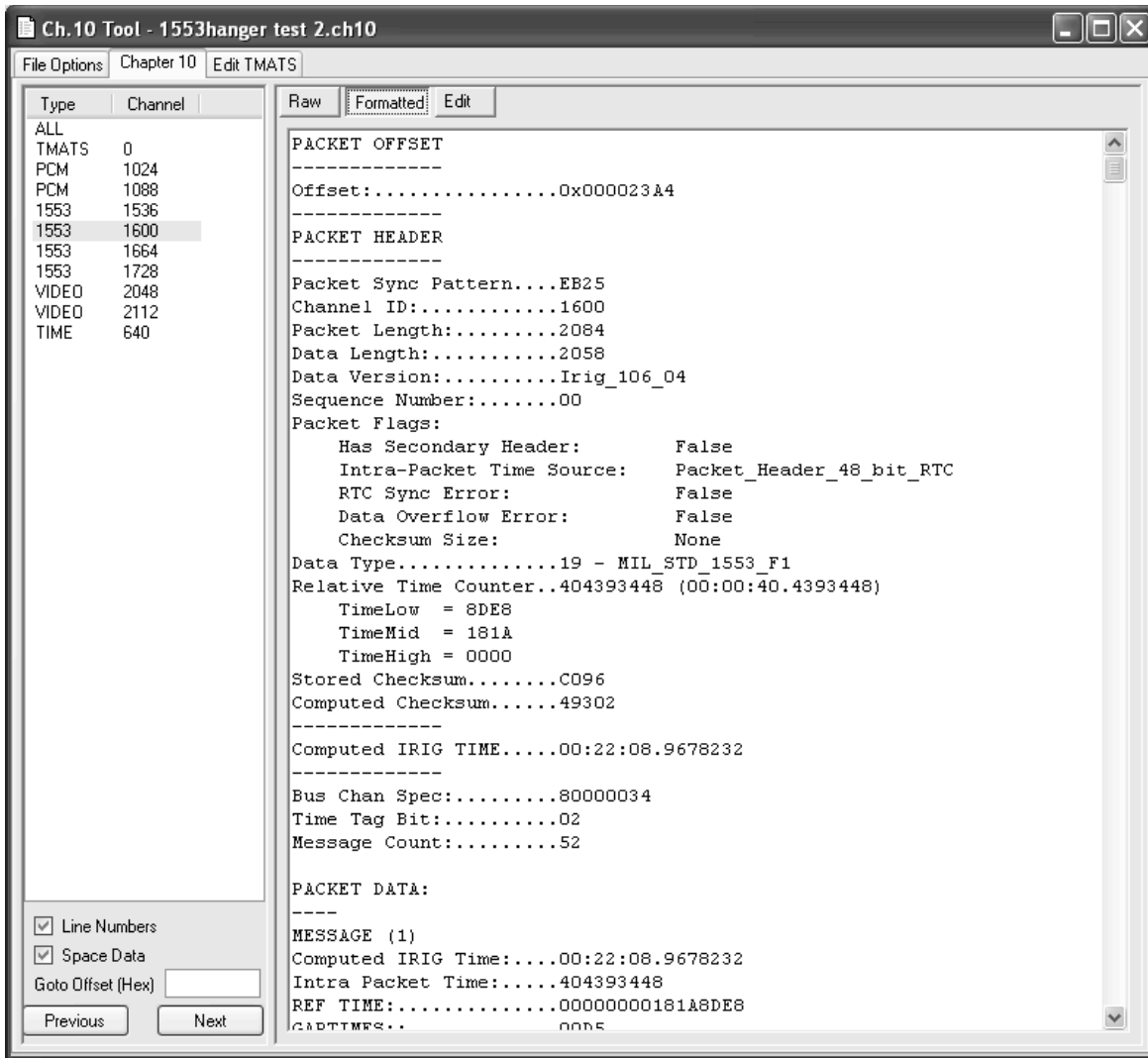


Figure 3. An example of the Chapter 10 Tool Packet Viewer Graphical User Interface: Formatted data.

Chapter 10 data files can range in sizes from kilobytes to gigabytes, thus, each packet will be viewed or edited individually. This method ensures a more efficient load time for viewing data. Indexing a duplicate file will also decrease processing time by avoiding the need to repetitively read through the entire file. When the validation of the Chapter 10 data file has been accomplished, the user will have the option to automatically correct errors. The user can jump to any specific packet within the Packet Viewer.

THE TELEMETRY ATTRIBUTES TRANSFER STANDARD EDITOR

The TMATS editor user interface, shown in Figure 4, is dynamically created at runtime based on a metadata dictionary derived from the TMATS eXtensible Markup Language (XML). The Chapter 9 Standard was used as a reference in determining the overall layout of the TMATS editor. When a Chapter 10 file is selected, a new tab is added to allow the user to edit, add, or

create TMATS data. A data tree is created from the metadata dictionary and grouped on the “Recorder/Reproducer” or “Transmission” Attributes. At runtime, the TMATS editing form is populated by creating control widgets based on the defined data type for each attribute (for example, text, number, selection list, et cetera). Each control widget is populated with the value from the Chapter 10 data file; controls are left blank when no matches are found. This method allows the user to add additional TMATS attributes as needed.

Ch.10 Tool - 1553hanger test 2.ch10

File Options Chapter 10 Edit TMATS

Tmats

- TransmissionAttributes
- RecorderReproducerAttributes
 - DS_TTC_LABhangerst1
 - MultiplexModulationGroupType

Program Name TTC_LABhangerst1

Test Item None

Tmats File Name

Revision Level 05

Origination Date 11-20-2007

Revision

Number 1

Date 01-02-2008

Update

Number

Date

Test Number

Number Of PO Cs

	Name	Agency	Address	Telephone
▶				
*				

Data Source

	Data Source ID	Data Source Type	Comments	Data Source ID
▶	DS_TTC_LABhangerst1	Other		
*				

Figure 4. An example of the Chapter 10 Tool Telemetry Attributes Transfer Standard Editor Graphical User Interface.

The difficulties in dealing with raw TMATS attributes are the various interpretations of the Chapter 9 Standard and the possibility of typographical errors within the selected TMATS file, which can lead to data being inadvertently excluded or an error occurring during the read process preventing the TMATS editor from displaying the contents of the file. If such errors are discovered, they are recorded in an error log; the user must correct any found errors before continuing to load the Chapter 10 file into the TMATS editor.

DESIGN EXPERIENCES AND PROBLEMS

Writing software for the Chapter 10 data file specification is relatively straightforward. Chapter 10 data structures are easily modeled using C-language structure types, and the little-endian nature of the standard is well-suited for Intel[®]-based (Intel Corporation, Santa Clara, California) computers running Microsoft Windows[®] (Microsoft Corporation). One challenge was modeling the Relative Time Counter; it was discovered to consist of three 16-bit, little-endian words - a fact that was not made clear in the Chapter 10 Standard.

The software architecture is written to simultaneously process multiple Chapter 10 channels on separate threads. Blocking queues are utilized to allow each channel to be checked for proper time alignment and packet order, and transformations can be applied to the data during the read/write process. Reading and writing Chapter 10 files is not difficult in and of itself; what is difficult is diagnosing and correcting data integrity problems. Data integrity problems can be highly domain-specific, thus, the software design allows for data processing plug-ins that can be written to address these problems on a case-by-case basis.

THE TELEMETRY ATTRIBUTES TRANSFER STANDARD MODULE

Writing software that follows the TMATS Chapter 9 Standard is not straightforward. Reading the TMATS information accurately is essential to properly interpreting the Chapter 10 data, especially with respect to PCM frame layout. The structure of TMATS data is simply a key-value data store, but the TMATS software must understand Group Relationships and the hierarchical, tree-like nature of the TMATS standard itself.

Due to the hierarchical nature of TMATS, recursive data structures are used extensively to model the TMATS data and give it some shape and context. This process allows the inclusion of additional embedded metadata, such as field descriptions, data length constraints and value lists. This metadata can be used to make the TMAT editor smarter and to assist the user in editing the TMATS information. The fields in the TMATS editor are dynamically generated from this metadata, which greatly reduces the development time for creating a user interface.

Since any TMATS attribute can appear anywhere in the Chapter 10 file, hash table indexes are used to allow rapid look-up of any attribute by group, group index, and code, ensuring adequate performance during reads. There is no guarantee that any given TMATS file has valid data, so a text editor is provided that can highlight potential data integrity problems in the TMATS data and which allows the user to visually correct these problems before the data are read into the main TMATS editor proper.

THE PULSE CODE MODULATION DATA TYPE MODULE

The PCM frame structure is defined in the “P Group” of the TMATS data packet. Each frame is identified by a frame synchronization (sync) pattern, marking the start of a frame. The TMATS P group identifies the sync bit pattern and the size of each PCM frame, so it is expected to see a

new frame sync pattern at the byte location identified by [start of frame] + [frame size]. If this does not occur, the current frame is considered invalid or corrupted. The software must then scan for the next frame sync pattern in the byte stream and confirm the presence of the next subsequent frame sync pattern in order to identify the next valid frame of PCM data.

Valid frames of data are passed through to the module interface where they can be processed by a custom module. Data frames can be visualized in an editor display driven by the TMATS frame definition; editing of individual bytes or frame values is possible.

ETHERNET DATA TYPE MODULE

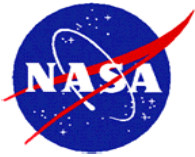
The Ethernet data type module detects and corrects errors and provides access to the information that is contained within each Ethernet packet. The module is planned to provide more capabilities specific to Ethernet data packets, such as filtering the data by a particular source or destination address.

The Ethernet data type module processes one Chapter 10 packet at a time, checking for errors, logging errors as they are found, and then passing the Chapter 10 packet back out of the module to the blocking queue.

As part of the ability to identify and isolate problems, or as a means of filtering data, the application provides access to the 48-bit destination media access control (MAC) address, the 48-bit source MAC address, the 16-bit Ethernet type (or length) value, and a 32-bit frame check sequence calculated with a CRC. The application examines the Ethernet type field to determine if the data represents a Data Link Level protocol type or if it contains the length of the Ethernet frame. These structures are written into arrays in the application in order to make it simpler to deal with the big-endian byte structure of Ethernet packets and the little-endian structure of Chapter 10 file specification.

CONCLUSION

While the first release is scheduled for Fall 2011, the development of the *Chapter 10 Tools* application is intended to evolve over time. The first release will include a Packet Viewer and support for Telemetry Attributes Transfer Standard and pulse code modulation data types. Subsequent releases will incorporate MIL-STD-1553, ARINC-429, and Ethernet data types. The intent is to create an easy-to-use application by eventually integrating the error correction intelligence into the application itself. Experience with the various Chapter 10 recorders should eventually yield patterns indicating how different manufacturers are interpreting the Standard and, by reading the name of the manufacturer in the Telemetry Attributes Transfer Standard portion of the Chapter 10 file, the application could presumably configure itself to automatically set up for specific types of errors and corrections.



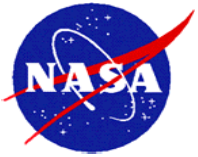
THE WESTERN AERONAUTICAL TEST RANGE

CHAPTER 10 TOOLS

Arcata Associates, Inc
Jack Sheldon (Primary Author)
Robert Downing
Robert Harvey
April Norcross

NASA Dryden
Kevin Knudtson
Alice Park (Intern)

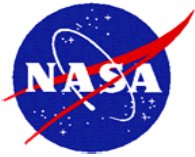
Oct. 27, 2011



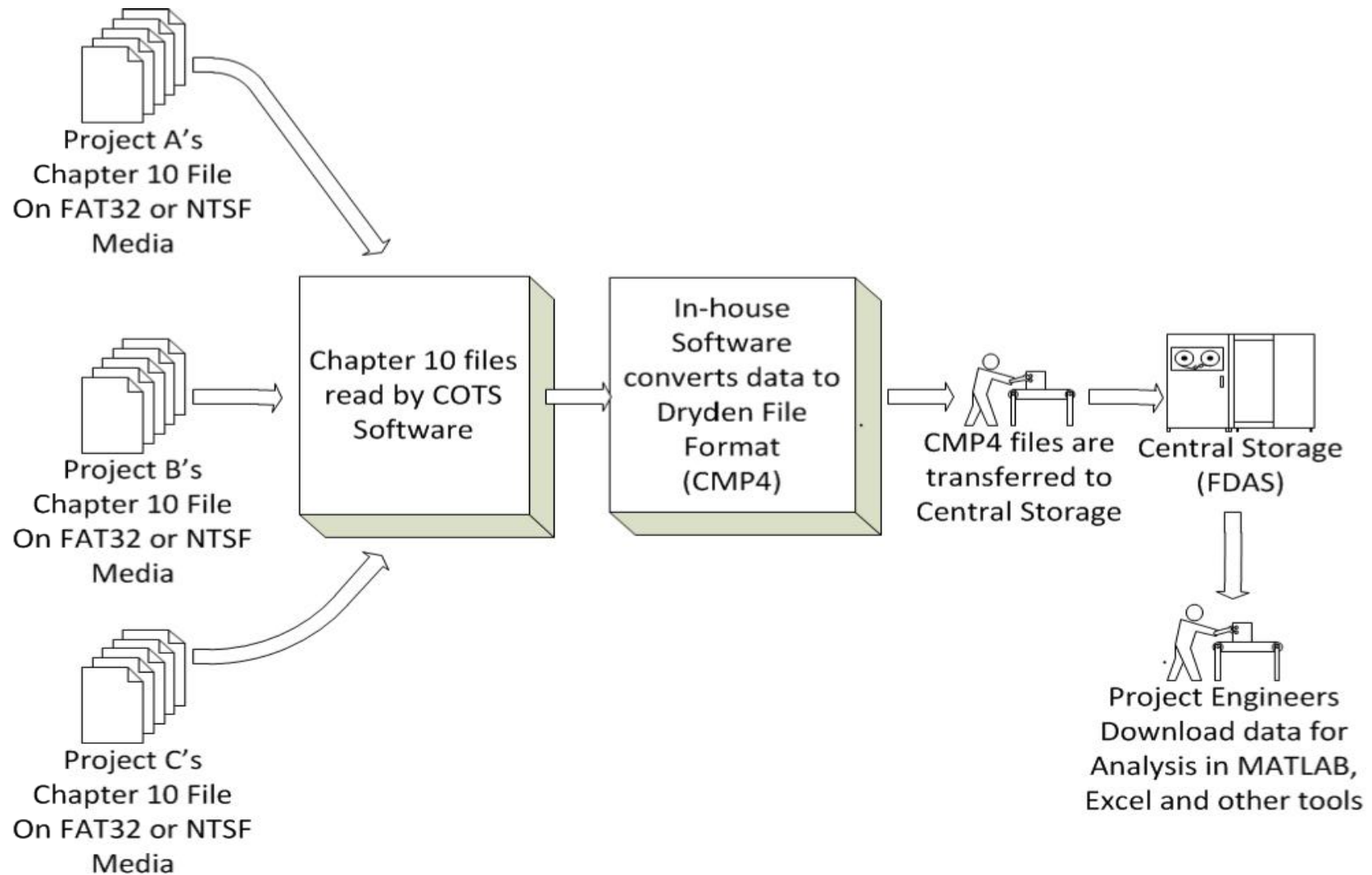
Agenda

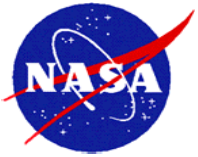


- NASA Dryden's Post-Flight Processing Environment
- Challenges with Processing Chapter 10 Data
- *Chapter 10 Tools* Application
- Demonstration Application & Questions



NASA Dryden's Post-Flight Processing Environment

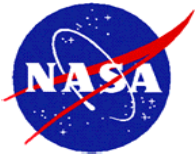




Challenges with Processing Chapter 10 Data



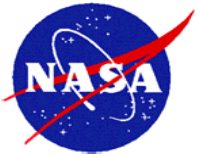
- The commercial software used to read the Chapter 10 files is unable to process every file
 - The software identifies an error and stops processing or the software stops processing with no error identified
- Most Common Problems
 - Time
 - Structure
 - TMATS



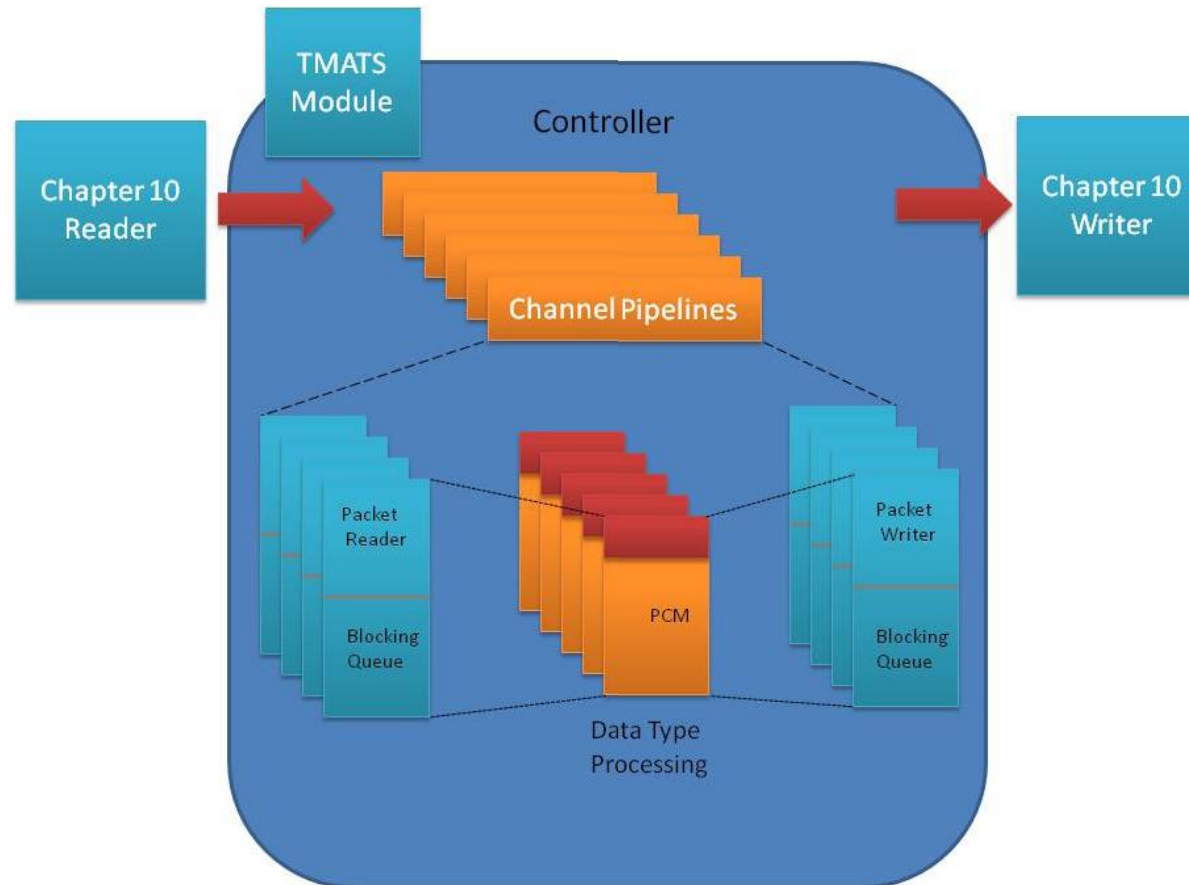
Chapter 10 Tools Application

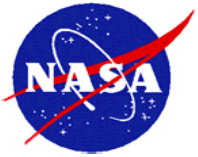


- Purpose for Chapter 10 Tools application
 - Create Chapter 10 files WATR can process
- Features within the Chapter 10 tools
 - Flexible - Plug-in architecture
 - Easy to use - Logical workflow
 - Productive - Chapter 10 file no longer a black box
 - Speed - Process big files fast
 - Capabilities
 - Chapter 10 packet validation
 - TMATS viewer/editor
 - Packet viewer/editor
 - Error detection (TMATS, PCM)
 - User specified automatic detection/correction (time)
 - Some automatic detection/correction (headers and trailers)
 - Output data into different formats, such as Matlab and CSV
 - Output new Chapter 10 files by time slices and/or data channels



Chapter 10 Tools Architecture





Demonstration and Questions